

REMARKS/ARGUMENTS

Prior to entry of this amendment, the application included claims 1-21. Claims 7 and 15 have been amended. No claims have been cancelled or added. Therefore, claims 1-21 remain present for examination.

Claim Objections

In the Office Action, claims 7 and 15 are objected to because of minor informalities. Specifically, that the phrases are considered to read as -- component [[registry]] repository -- as previously recited in independent claims 1 and 10, respectively.

Claims 7 and 15 have been amended to obviate the informality and Applicants request withdrawal of the objection.

Rejections Under 35 U.S.C. § 102

The Office Action rejected claims 1-21 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Publication No. 2004/0040014 A1 to Ball ("Ball"). The patent office is charged with putting forth a *prima facie* showing of anticipation. Applicants believe a *prima facie* case of anticipation has not been properly set forth in the Office Action. The basic test is excerpted below:

“[For] anticipation under 35 U.S.C. 102, the reference must teach every aspect of the claimed invention either explicitly or impliedly.” See MPEP §706.02, Original Eighth Edition, August, 2001, Latest Revision August 2006.

Applicants believe the rejection has flaws with the above test for anticipation. Notably, the Ball reference fails to teach each and every aspect of the claimed invention.

Applicants submit that Ball discloses a system used to analyze software applications during coding and development of the application. (See Ball at [0014].) The system is configured to use testing tools to generate metrics on compiled software. (See Ball at [0030] and [0039].) Testing tools use static analysis to determine the attributes of compiled code. (See

Ball at [0002] - [0003].) Testing tools are different from other types of software testing conducted on the runtime software. (*Compare* Ball [0001] and [0002].) Indeed, the system of Ball provides metrics during software development. (*See* Ball at [0045].)

In contrast, independent claim 1 recites, in pertinent part, “monitoring the application component *in the IDE runtime environment*.” (emphasis provided). The Examiner asserts that Ball, at [0035] - [0036], discloses monitoring the application in the IDE runtime environment. (*See* Office Action, page 4.) However, a review of that passage (as well as the remainder of Ball) fails to reveal any disclosure of monitoring the application in any runtime environment. Instead, ¶¶ [0009], discloses “Testing tools are often used to determine values of metrics.” Testing tools are “another aspect of software testing,” (*see* Ball at [0002]) from the runtime testing, such as black box testing, stress and load testing, etc. (*Compare* Ball [0001] and [0002].) Applicants submit that the metrics generated in Ball are during development of the software and not during the runtime testing of software. Specifically, Ball discloses:

Provision is made for timely feedback to the user of the development tool regarding values of software metrics as applied to software applications *being developed by the user*. For example, instead of a separate metrics report sent to the management team at designated points (e.g., mid-release cycle) in the software development cycle, timely, prompt feedback is delivered to each software developer via a modification of the graphical display of an IDE. Furthermore, persistent display of metrics feedback icons using the IDE, which the developer usually sees on a daily basis, may provide improved follow-through on correction of software problems, because the developer is provided with reminders via the feedback icons *during development of software*. Thus, using the present invention, quality of the software application may be enhanced and cost of production of the software may be lowered. *Provision is made to lower cost of software development because correction of software problems indicated by the metrics feedback icons is made early in the development cycle.*

Ball at [0045] (*emphasis added*). Providing metrics during development of the software is very different than “monitoring the application component *in the IDE runtime environment*.”

Moreover, the assertion that Ball teaches the use of testing tools in a development environment is bolstered by the metrics obtained. Ball recites that the metric obtained by the development tool “include Depth of Inheritance Tree (DIT), Number of Children (NOC),

Coupling Between Objects (CBO), Weighted Methods per Class (WMC), Response for Class (RFC), Message Passing Coupling (MPC) . . .” (see Ball at [0030], [0031] and [0010]). These metrics do not pertain to the operation of software in a runtime environment. Rather, these metrics described in Ball pertain to “‘good code’ [which] is considered for complexity, extensibility, modularity, readability, etc.” (See Ball at [0009]), not for runtime operation. Consequently, the system of Ball and the systems and methods defined in the present claims may be used in concert rather than used as alternatives. That is, the system of Ball can be used to create better code during development, while the systems and methods defined in the present claims may be used to determine how well the code executes in operation in a runtime environment.

Consequently, Ball’s development tool performs a completely different function from claim 1’s “monitoring the application component in the IDE runtime environment.” Moreover, because Ball’s development tool provides different metrics by using testing tools in a development environment, e.g., code scanning, there is no way the system of Ball could be modified to function to “monitoring the application component in the IDE runtime environment,” as recited by claim 1. As such, Applicants respectfully submit that Ball fails to anticipate each and every element of independent claim 1 and claim 1, therefore, is allowable over Ball.

Claim 9, like claim 1, recites “monitor the application component in the IDE runtime environment.” Claim 9, therefore, is allowable for at least similar reasons. Claim 10 recites, *inter alia*, “an IDE runtime environment configured to open an application component and monitor operation of the application component.” For at least the reasons noted above, Ball does not provide a system to monitor an application in a runtime environment. Claim 10, therefore, is allowable over Ball as well.

Dependent claims 2-8 and 11-21 each ultimately depend from either claim 1, claim 9, or claim 10, and therefore are allowable over Ball at least by virtue of their dependence from allowable base claims.

Appl. No. 10/751,333
Amdt. dated: December 3, 2007
Reply to Office Action of: August 2, 2007

PATENT

CONCLUSION

In view of the foregoing, the applicants believe all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 303-571-4000.

Respectfully submitted,



Tadd F. Wilson
Reg. No. 54,544

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 303-571-4000 (Denver office)
Fax: 303-571-4321 (Denver office)

TFW:slb
61151063 v1